TRANSPORTATION ANALYSIS SIMULATION SYSTEM
(TRANSIMS)

VERSION 1.0

Input Editor Subsystem for IOC-1

B. W. Bush
Energy and Environment Analysis Group
Los Alamos National Laboratory

Modified by P. Medvick

March 1998

[Tab 7]

Contents

1. INTRODUCTION

The TRANSIMS input editor separates the user from the lower-level layers of TRANSIMS software involved with data management. Figure 1 shows the position of the input editor within the TRANSIMS software architecture.

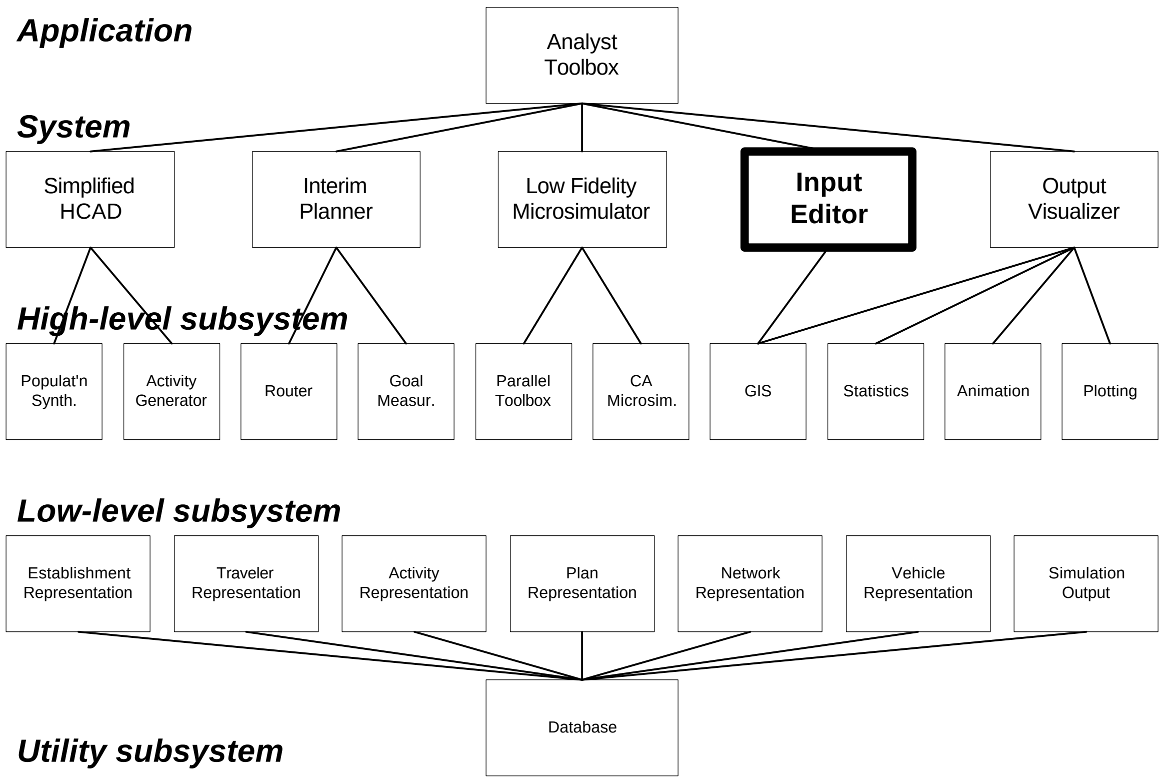

Figure 1: Location of the Input Editor System within the TRANSIMS Software Architecture

The input editor provides a variety of data editing functions. It accepts standard file formats (dBASE, delimited text, and ArcInfo), and it allows form-, map- and spreadsheet-style editing of data tables. A graphical intersection viewer allows a user to visualize the allowed lane-to-lane vehicle movements in the various operational phases of a traffic signal. Tools support the validation of network data so that problems in network data (errors, anomalies, and inconsistencies) are quickly identified. Additional facilities allow the creation of simulation output specification tables. The TRANSIMS Analyst Interface (TAI) may start the input editor either specifically for creating output specification tables or as a more general-purpose network data viewing and manipulation subsystem.

The input editor is integrated into the ArcView geographic information system (GIS). It seamlessly connects to the ArcInfo product and relational databases. One can also customize or extend the input editor using the Avenue programming language.

The input editor can run under several different modes. Access through the TAI is to the output specification and analyst modes. The severely limited menus in the output specification mode permit viewing and selecting links or nodes for the output specification file, and writing link or node ids to a file. The general usage through the TAI interface (network button after a network is selected) permits viewing of networks and provides TRANSIMS-specific tools for viewing intersections and tools for data validation. The developers mode, which is not supported in this release, has more capabilities, based primarily on interaction with an Oracle database.

2. CONCEPTS

ArcView [ES 96a] primarily uses the Shape and dBASE file formats for data storage. The shape file format is a portable open standard for files that contain geographic data. Shape files are typically paired with dBASE files containing the non-geographic attributes of the geographic objects in the shape file. The dBASE format is a portable industry standard for files that contain tabular data. Both formats can easily be imported or exported between a variety of commercial products.

Note: ArcView is not able to handle capital letters in directory names used within Input Editor.

3. USAGE

3.1 Setup

The file, $TRANSIMS_HOME/data/ined/default.apr, must be located, or linked to, from your home directory.

3.2 Menus

The input editor provides several custom menus (see Table 1), in addition to the standard ArcView menus, that allow access to the input editor functions. Three menus, described in Table 2 through Table 4, are available with the standard ArcView startup with one argument (arcview project-file-name). The last two menus, Table 5 and Table 6, are for output specification file creation. They are accessible either via the TAI in **Microsimulation Setup/Output Collection** or by starting ArcView with two arguments (refer to the tutorial Creating Output Specification Files).

3.2.1 Analyst Mode Menus

Table 1: Input Editor Menus

Menu	Description	Availability
Tables	Renames network tables	When the Project window is active and in analyst mode
Validate	Validates network tables	When the Project window is active and in analyst mode
Intersection	Views allowed movements at an intersection	When a view window is active and in analyst mode
Display Highlights from File(s)	Highlights links and/or nodes listed in a file(s)	A View window is active and in output specification mode (initiated by TAI or by two input arguments at startup)
Save Highlights from File(s)	Saves highlighted link and/or node ids from view into file(s)	A View window is active and in output specification mode (initiated by TAI or by two input arguments at startup)

Table 2: Tables Menu Items (available when the Project window is active)

Menu Item	Description
Set Nodes	Specifies the node table to use for analysis
Set Links	Specifies the link table to use for analysis
Set Pocket Lanes	Specifies the pocket lane table to use for analysis
Set Lanes	Specifies the lane table to use for analysis
Set Connectivity	Specifies the lane connectivity table to use for analysis
Set Unsignalized Nodes	Specifies the unsignalized node table to use for analysis
Set Timing Plans	Specifies the timing plan table to use for analysis
Set Signalized Nodes	Specifies the signalized node table to use for analysis
Set Phasing Plans	Specifies the phasing plan table to use for analysis
Set Parking	Specifies the parking table to use for analysis
Set Study Area	Specifies the study area table to use for analysis

Table 3: Validate Menu Items (available when the Project window is active)

Menu Item	Description
Nodes	Validates the contents of the node table
Links	Validates the contents of the link table
Pocket Lanes	Validates the contents of the pocket lane table
Connectivity	Validates the contents of the lane connectivity table
Unsignalized Nodes	Validates the contents of the unsignalized node table
Timing Plans	Validates the contents of the timing plan table
Signalized Nodes	Validates the contents of the signalized node table
Phasing Plans	Validates the contents of the phasing plan table
Parking	Validates the contents of the parking table
Study Area Link	Validates the contents of the study area table
All	Validates all of the above tables in the proper order

Table 4: Intersection Menu Items (available when a View window is active)

Menu Item	Description
Set Intersection	Makes the currently selected node the current intersection
Show Intersection	Shows the current intersection
Set Protections/Signs	Specifies the protection and sign codes to be viewed for the current intersection
Show Protections/Signs	Shows the protection and sign codes to be viewed for the current intersection
Set Phase	Specifies the phase to be viewed for the current intersection
Show Phase	Shows the phase to be viewed for the current intersection
Next Phase	Specifies the phase to be viewed for the current intersection as one of the phases immediately after the current phase
Previous Phase	Specifies the phase to be viewed for the current intersection as one of the phases immediately before the current phase
Show Incoming Lanes	Selects the lanes from which movements are allowed to the currently selected lanes
Show Outgoing Lanes	Selects the lanes to which movements are allowed from the currently selected lanes
Save Lane Selection	Remembers the currently selected lanes
Restore Lane Selection	Restores the lanes selection to the last one saved

3.2.2 Output Specification Menus

These menus are available when a View window is active and ArcView was started either with two arguments (arcview project-name-file output-spec-info-file) or from the TAI for creation of a link or node output specification file. The visibility of the menu items is predetermined by the type of data being selected: node or link.

Table 5: Display Highlights from a File

Menu Item	Description
Display Nodes	Highlights the nodes listed in the input node file
Display Links	Highlights the links listed in the input link file

Table 6: Save Highlights to a File

Menu Item	Description
Save Nodes	Saves the highlighted node ids to the node output specification file
Save Links	Saves the highlighted link ids to the link output specification file

3.3 Tutorials

3.3.1 Network Data Validation

Table 7 outlines the procedure for validating network data. Before starting the validation process, all of the data should be in dBASE (.dbf), INFO, or delimited text (.txt) with commas as the delimiter format. In the tutorial, *.dbf will be used to represent the three possible formats.

The steps in **Table 7** must be performed in sequence.

☛: The **Tables** and **Validate** drop-down menus are accessible only when the Project window is active.

Table 7: Network Data Validation

Step	Function	Action
1.	Validate the node table	♦ Make the Project window active. Note: if the node table is not already in the input editor and named "Nodes", perform the actions in the four clear bullets; otherwise, skip to the next filled bullet (validate). ◊ Make sure the node table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf node file while in the **Add Table** window. ◊ Select **Tables→Set Nodes** to set the node table name. ♦ Select **Validate→Nodes** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The IDs have legal values. • The IDs are unique. ♦ Fix any errors that were detected ♦ Re-run the validation script until no errors are found.

Step	Function	Action
2.	Validate the link table	◊ Make sure the link table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf link file while in the **Add Table** window. ◊ Select **Tables→Set Links** to set the link table name. ♦ Select **Validate→Links** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The IDs have legal values. • The IDs are unique. • The nodes at the endpoints exist. • All nodes have at least one incoming and one outgoing link. • The values of all of the fields are valid. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.
3.	Validate the pocket lane table	◊ Make sure the pocket lane table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select your .dbf pocket lane file while in the **Add Table** window. ◊ Select **Tables→Set Pocket Lanes** to set the pocket lane table name. ♦ Select **Validate→Pocket Lanes** to run the validation script. The log file will contain a description of the problems found. The corresponding records for the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The IDs have legal values. • The IDs are unique. • The styles are valid. • The node and link references are correct. • The lane number is that of a valid pocket lane. • The offset and length are consistent with the setbacks and length of the link. • All of the pocket lanes specified in the link table are present. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.

Step	Function	Action
4.	Validate the lane connectivity table	◊ Make sure the connectivity table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select your .dbf connectivity file while in the **Add Table** window. ◊ Select **Tables→Set Connectivity** to set the lane connectivity table name. ♦ Select **Validate→Connectivity** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The node, link, and lane references are correct. • Each lane has at least one incoming and one outgoing lane connection. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.
5.	Validate the unsignalized node table	◊ Make sure the unsignalized node table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf unsignalized node file while in the **Add Table** window. ◊ Select **Tables→Set Unsignalized Nodes** to set the unsignalized node table name. ♦ Select **Validate→Unsignalized Nodes** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The signs are valid. • The node and link references are correct. • Each incoming link is controlled. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.

Step	Function	Action
6.	Validate the timing plan table	◊ Make sure the timing plan table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf timing plan file while in the **Add Table** window. ◊ Select **Tables→Set Timing Plans** to set the timing plan table name. ♦ Select **Validate→Timing Plans** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The plans and phase values are legal. • The (plan, phase) pairs are duplicated. • The time values are legal and consistent. • The phase sequence references existent phases. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.
7.	Validate the signalized node table	◊ Make sure the signalized node table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf signalized node file while in the **Add Table** window. ◊ Select **Tables→Set Signalized Nodes** to set the signalized node table name. ♦ Select **Validate→Signalized Nodes** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The types are valid. • The node references are correct. • Each node has at most one signalized control (or zero if there is an unsignalized control at the node). • The plan references are correct. • Each node has either a signalized or unsignalized control. • All plans are used. • The start times are valid. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.

Step	Function	Action
8.	Validate the phasing plan table	◊ Make sure the phasing plan table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf phasing plan file while in the **Add Table** window. ◊ Select **Tables→Set Phasing Plans** to set the phasing plan table name. ♦ Select **Validate→Phasing Plans** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The protections are valid. • The plan, phase, node, and link references are correct. • Each incoming and outgoing link is controlled. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.
9.	Validate the parking table	◊ Make sure the parking table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf parking file while in the **Add Table** window. ◊ Select **Tables→Set Parking** to set the parking table name. ♦ Select **Validate→Parking** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed: • The field names and types (character vs. numeric) are correct. • The styles, range of capacity, and generic fields are valid. • The node and link references are correct. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.

Step	Function	Action
10.	Validate the study area table	◊ Make sure the study area table is in *.dbf format. ◊ Select **Project→Add Table**. ◊ Select the .dbf study area file. ◊ Select **Tables→Set Study Area** to set the study area table name. ♦ Select **Validate→Study Area Link** to run the validation script. The log file will contain a description of the problems found. The corresponding records of the tables involved will be selected. The following checks are performed. • The field names and types (character vs. numeric) are correct. • The id and buffer fields are valid. • The link references are correct. ♦ Fix any errors that were detected. ♦ Re-run the validation script until no errors are found.

3.3.2 Viewing Intersections

Table 8 provides the tutorial for viewing intersections. Before beginning the viewing, all of the network data tables must be loaded as specified in the network data validation tutorial (see Table 7). Also, a view must contain themes named Nodes, Links, and Lanes. For convenience, menu commands are available for saving (**Intersection→Save Lane Selection**) and restoring (**Intersection→Restore Lane Selection**) the selection state.

Table 8: Viewing Intersections

Step	Function	Action
1.	Select the intersection	♦ In the View window, select the node containing the intersection of interest. This may be a signalized or an unsignalized intersection. Select **Intersection→Set Intersection** to set this selected node as the current intersection to be viewed.
2.	Select the phase	♦ Select **Intersection→Set Phase** to choose the phase to be viewed at this intersection. (Use **Intersection→Next Phase** and **Intersection→Previous Phase** to move to adjacent phases.)

3.	Select the protections or signs of interest	♦ If only certain protections or signs are of interest, select **Intersection→Set Protections/Signs** to specify which should be considered. Select the ones of interest in each category:
		◊ Protection codes:
		P = Protected
		U = Unprotected
		◊ Sign codes:
		S = Stop light
		Y = Yield sign
		N = None

4.	Select the lanes of interest and view the allowed movements	♦ In the View window, select one or more lanes for which the allowed movements are to be viewed. Note: the lanes need to be visible and active to do this. If these are incoming lanes, select **Intersection→Show Outgoing Lanes** to see to which lanes these lead. If they are outgoing lanes, select **Intersection→Show Incoming Lanes** to see which lanes lead to them.

3.3.3 Creating Output Specification Files

Table 9 outlines the procedure for storing highlighted link and node ids in output specification data files.

Table 9: Creating Output Specification Files

Step	Function	Action
1.	Create files for startup	♦ One to three files must be created. Note that if you are working from the TAI, these files are created automatically; skip to Step 2. ♦ The required file (FILE1) contains three lines of information and no blank lines: ◊ Line 1 has the keyword *Link* or *Node*, followed by a space and then either 'link' or 'node'. ◊ Line 2 has the keyword *LinkFile*, followed by a space and then the full path to the file for link ids. ◊ Line 3 has the keyword *NodeFile*, followed by the full path to the file for node ids. Example: if no ids are to be displayed and no selected ids are to be saved, '/dev/null' may be used as the file name. ♦ To select a set of links, have 'link' as the second word in line 1 of FILE1 and create a file for the link ids. The link ids listed in the file will be highlighted at startup. No blank lines may be in the link id file. ♦ To select a set of nodes, have 'node' as the second word in line one of FILE1 and create a file for the node ids. The node ids listed in the file will be highlighted at startup. No blank lines may be in the node id file.
2.	Start up ArcView in output specification mode	♦ If starting by hand, type `arcview <path>project_file.apr FILE1`. ♦ Or, in the TAI, after the network, activities, and plans are set, select **Microsim Setup→Output Collection**. The **Start ArcView** check boxes initiate the ArcView call.
3.	If links are to be selected	♦ With the View window active, choose the selection tool, and make sure the "Links" theme is active. ♦ Select the links on which to collect microsimulation data. They will become highlighted.

Step	Function	Action
4.	If nodes are to be selected	♦ With the View window active, choose the selection tool, and make sure the “Nodes” theme is active. ♦ Select the nodes on which to collect microsimulation data. They will become highlighted.
5.	Save the highlighted ids to a file	♦ Select **Save Highlights to File(s)**. The appropriate **Save Links** or **Save Nodes** will be displayed. The ids will be stored in the file specified in FILE1.
6.	Exit mode	♦ Select **File→Exit** to return to the TAI or your operating system.

4. USAGE DEVELOPERS

Developers mode, although not supported in this release of the TRANSIMS software, can provide access to additional features of the input editor. The mode menu is accessible from the menu bar for the Project document, and when visible, provides the ability to toggle to developers mode. Interaction with an Oracle database and the ability to create shape files are added capabilities of the developers mode.

5. REFERENCES

[ES 96a] *ArcView GIS: The Geographic Information System for Everyone,* (Readlands, California: Environmental Systems Research Institute, 1996).